The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Monocular Vision SLAM for Indoor Aerial Vehicles

Koray Çelik, Soon-Jo Chung, Matthew Clausman, and Arun K. Somani

***Abstract*— This paper presents a novel indoor navigation and ranging strategy by using a monocular camera. The proposed algorithms are integrated with simultaneous localization and mapping (SLAM) with a focus on indoor aerial vehicle applications. We experimentally validate the proposed algorithms by using a fully self-contained micro aerial vehicle (MAV) with on-board image processing and SLAM capabilities. The range measurement strategy is inspired by the key adaptive mechanisms for depth perception and pattern recognition found in humans and intelligent animals. The navigation strategy assumes an unknown, GPS-denied environment, which is representable via corner-like feature points and straight architectural lines. Experimental results show that the system is only limited by the capabilities of the camera and the availability of good corners.**

I. Introduction

The foreseeable future of intelligence, surveillance and reconnaissance missions will involve GPS-denied environments. An MAV with vision based on-line simultaneous localization and mapping (SLAM) capabilities can pave the way for an ultimate GPS-free navigation tool for both urban outdoors and architectural indoors. While the severe payload constraints of MAVs prevent the use of conventional sensors such as laser range-finders, the astounding information-to-weight ratio of vision makes it worthwhile to investigate. However, vision captures the geometry of its surrounding environment indirectly through photometric effects. In order to solve the depth problem, the literature resorted to various methods such as the Scheimpflug principle, structure from motion, optical flow, and stereo vision. None of these have a potential for on-line SLAM applications with reasonable computation as well as robustness, with respect to a wide range of depths, and with reasonable computation on a *small flying MAV*. For example, the ocular separation of stereo vision significantly limits its practical application and useful range. Parabolic and panoramic cameras [1] are heavy, and thus, better suited for ground vehicles [2]. The use of moving lenses for monocular depth extraction [3] is not applicable to SLAM since this method cannot focus at multiple depths at once. Optical flow sensors [4], [5] require incessant motion and hence becomes less useful in a hovering MAV, while image patches obtained are too ambiguous for the landmark association procedure for SLAM.

This paper presents one of the smallest fully self-contained autonomous helicopters equipped with sophisticated on-board image processing (see Fig. 9 and Section V for details). Our approach accounts for how a human perceives depth via monocular visual cues such as line perspectives, relative height, texture gradient, and motion parallax. We then integrate this ranging technique with SLAM to achieve autonomous indoor navigation of an MAV. Although we emphasize that our real-time algorithms are validated by a small fully self-contained aerial vehicle, they can be applied to any mobile platform with known height.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors which show the experimental results of the paper. This material is 6.9 MB in size.

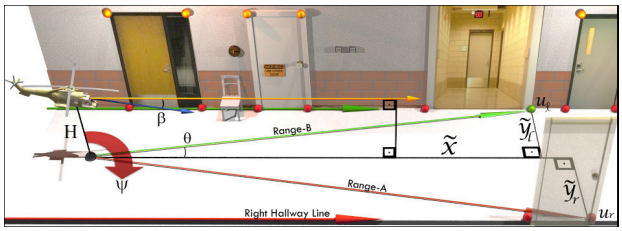

Fig. 1. A three dimensional representation of the corridor showing line perspectives and corner-like features.

A. Related Work on Vision-based SLAM

We emphasize that prior works, which are otherwise excellent, are not directly applicable to our particular application. Vision research has particularly concentrated on Structure from Motion (SFM) to produce a reconstruction of the camera trajectory and scene structure [6], [7], [8]. This approach may be suitable for solving the offline-SLAM problem in small image sets. However, an automatic analysis of the recorded footage from a completed mission cannot scale to a consistent localization over arbitrarily long sequences in real-time.

Extended Kalman Filter (EKF) based approaches to probabilistic vision based SLAM, such as the elegant method of MonoSLAM [9], are excellent for applications requiring precise and repeatable localization within the immediate vicinity of a known, calibrated starting point. However, an MAV covers a very large unknown area in which the mission can start at any arbitrary location. A more recent work [10] presented a different approach to mitigate the issues involving long distances by means of map matching. However, the depth measurement is relative which would not provide reliable object avoidance for an agile flying MAV in a relatively narrow indoor environment, and the computational requirements are beyond reasonable limits for an MAV.

Global localization techniques such as Condensation-SLAM [11] show very successful localization performance. However, they require a full map to be provided to the robot a-priori. Azimuth learning based techniques such as

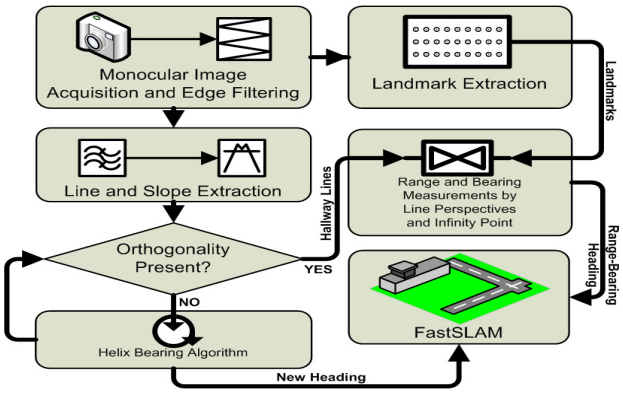

Fig. 2. Block diagram illustrating the operational steps of the monocular vision navigation and ranging at high level.

CognitiveSLAM [12] are parametric, and locations are centered on the robot which naturally becomes incompatible with ambiguous landmarks. Image registration based methods, such as [13], propose a different formulation of the vision-based SLAM problem based on motion, structure, and illumination parameters without first having to find feature correspondences. For a real-time implementation, however, a local optimization procedure is required, and there is a possibility of getting trapped in a local minimum. Further, without merging regions with a similar structure, the method becomes computationally intensive considering the limitations of MAVs. The structure extraction method [14] has its own limitations since an incorrect incorporation of points into higher level features will have an adverse effect on consistency. Higher level structures are purely constructed from the information contained in the map while there is an opportunity to combine the map with the camera readings. Further, these systems depend on a successful selection of thresholds which have a considerable impact on the system performance, thus limited to small scale maps.

B. Organization

This paper addresses the above shortcomings using a monocular camera of 1×2 inches in size and less than 2 ounces in mass. By exploiting the architectural orthogonality of the indoor environments, we introduce a novel method for monocular vision based SLAM by computing absolute range and bearing information without using active ranging sensors. More thorough algorithm formulations and newer experimental results with an MAV are discussed in this paper than in our prior conference articles [15], [16]. Section II explains the procedures for perception of world geometry as pre-requisites for SLAM. While a visual turn-sensing algorithm is introduced in Section III, SLAM formulations are provided in Section IV. Results of experimental validation as well as a description of the MAV hardware platform are presented in Section V. Figure 2 can be used as a guide to sections as well as to the process flow of our proposed method.

II. Problem and Algorithm Formulation

We propose a novel method to estimate the absolute depth of features using a monocular camera as a sole means of navigation. The only a-priori information required is the altitude above ground, and the only assumption made is that the landmarks are stationary. Altitude is measured in real-time via the on-board altimeter. We validate our results with time-varying altitude. It is also possible to operate this system on a fixed height device.

A. Landmark Extraction

No SLAM approach is a dependable solution without reliable landmarks. A landmark in the SLAM context is a conspicuous, distinguishing landscape feature marking a location. This definition is sufficient for SLAM, but not necessary. A minimal landmark can consist of range and bearing. To automate landmark extraction, we begin extracting prominent parts of the image that are more attractive than other parts in terms of energy. A corner makes a nice feature. But the wall itself is uniform and thus unlikely to attract a feature scanner. Landmarks in the real 3D world are distinctive whereas features exist on the 2D image plane and they are ambiguous. We select and convert qualifying features into landmarks as appropriate.

In our preliminary results [15], we have tried the Harris corner detection algorithm. However, due to its Markovian nature, the algorithm was not well suited for tracking agile motion; a feature detector, not an efficient feature tracker, as every frame is considered independently. Although in slow image sequences, this may provide a sparse and consistent set of corners due to its immunity to affine transformations and noise, we have obtained the best feature detection, and tracking performance from the algorithm proposed by Shi and Tomasi [18], which works by minimizing the dissimilarity between past images and the present image in a sequence. Features are chosen based on their monocular properties such as texture, dissimilarity, and convergence; sections of an image with large eigenvalues are considered "good" features; conceptually similar to the surface integration of the human vision system. However, this method cannot make an educated distinction between an useless feature and a potential landmark. That distinction is later performed by our proposed method, extracting a sparse set of reliable landmarks from a populated set of questionable features as described in Sections II-B and IV-A.

B. Line and Slope Extraction

For our range measurement approach to work, the architectural ground lines should be extracted. On an ideal, well-lit and well-contrasting hallway, ground lines are often obvious. However, on a monocular camera, the far end of a hallway appears too small on the image plane, and therefore is aliased. On a video feed, the corresponding ends of the hallway lines would translate randomly. Stochastic presence and absence of these perturbations result in lines that are inconsistent about their position. This in turn leads to noisy

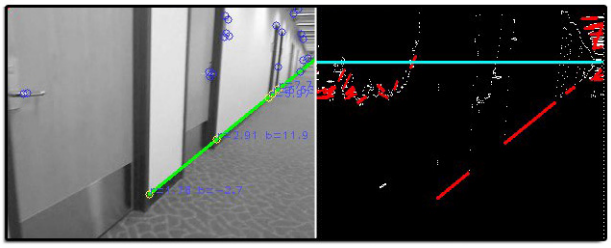

Fig. 3. Initial stages after filtering for line extraction, in which the line segments are being formed. The horizontal line across the image denotes the artificial horizon for the MAV.

slope measurements and eventually noisy landmarks. The construction should be an adaptive approach.

We begin the adaptive procedure by passing the image, I, through a discrete differentiation operator with more weight on the horizontal convolution, such as

$$I'_x = F_h * I, \text{ and } I'_y = F_v * I \tag{1}$$

where $*$ denotes the convolution operator, and F is a 3×3 kernel for horizontal and vertical derivative approximations. I'_x and I'_y are combined with weights whose ratio determine the range of angles through which edges will be filtered. This in effect returns a binary image plane, I', with potential edges that are more horizontal than vertical. It is possible to reverse this effect to detect other edges of interest, such as ceiling lines, or door frames. At this point, edges will disintegrate the more vertical they get (see Fig. 3 for an illustration). Application of the Hough Transform to I' will return all possible lines, automatically excluding discrete point sets, out of which it is possible to sort out lines with a finite slope $\phi \neq 0$ and curvature $\kappa = 0$. Nevertheless, this is an expensive operation to perform on a real-time video feed since the transform has to run over the entire frame. To improve the overall performance in terms of efficiency, we have investigated replacing Hough Transform with an algorithm that only runs on parts of I' which contain data. This approach begins by dividing I' into square blocks, $B_{x,y}$. Optimal block size is the smallest block that can still capture the texture elements in I'. Camera resolution and filtering methods used to obtain I' have a large effect on the resulting texture element structure. The blocks are sorted to bring the highest number of data points with the lowest entropy first, as this is a block most likely to contain lines. Blocks that are empty, or have a few scattered points in them, are excluded from further analysis. Entropy is the characteristic of an image patch that makes it more ambiguous, by means of disorder in a closed system. This assumes that disorder is more probable than order, and thereby, lower disorder has higher likelihood of containing an architectural feature.

The set of *candidate* blocks resulting at this point are to be searched for lines. Although a block B_n is a binary matrix, it can be thought as a coordinate system which contains a set of points (i.e., pixels) with (x, y) coordinates such that positive x is right, and positive y is down. Since we are more interested in lines that are more horizontal than vertical, it is safe to assume that the errors in the y values outweigh that of in the x values. Equation for a ground line is in the form $y = mx + b$, and the deviations of data points in the block from this *line* are, $d_i = y_i - (mx_i + b)$. Therefore, the most likely *line* is the one that is composed of data points that minimize the deviation such that $d_i^2 = (y_i - mx_i - b)^2$. Using determinants, the deviation can be obtained as in (2).

$$d_i = \begin{vmatrix} \sum(x_i^2) & \sum x_i \\ \sum x_i & i \end{vmatrix}, \quad m \times d_i = \begin{vmatrix} \sum(x_i.y_i) & \sum x_i \\ \sum y_i & i \end{vmatrix}$$
$$b \times d_i = \begin{vmatrix} \sum(x_i^2) & \sum(x_i.y_i) \\ \sum x_i & \sum y_i \end{vmatrix} \tag{2}$$

Since our range measurement methods depend on these lines, measurement noise in slopes has adverse effects on SLAM and should be minimized to prevent inflating the uncertainty. To reduce this noise, lines are cross-validated for the longest collinearity via pixel neighborhood based line extraction, in which the results obtained rely only on a local analysis. Their coherence is further improved using a post-processing step via exploiting the texture gradient. Note that this is also applicable to ceiling lines. Although ground lines (and ceiling lines, if applicable) are virtually parallel in the real world, on the image plane they intersect, and the horizontal coordinate of this intersection point is later used as a heading guide for the MAV. Features that happen to coincide with these lines are potential landmark candidates.

C. Range Measurements by the Infinity-Point Method

Inspired by [20], our monocular ranging algorithm attempts to learn from the human perception system, and accurately measures the absolute distance by integrating local patches of the ground information into a global surface reference frame. This new method, efficiently combined with the feature extraction method and SLAM algorithms, significantly differs from optical flows in that the depth measurement does not require a successive history of images.

Once features and both ground lines are detected, our range and bearing measurement strategy assumes that the height of the camera from the ground, H, is known a priori (see Fig. 1). This can be the altimeter reading of the MAV. The camera is pointed at the far end of the corridor, tilted down with an angle β. The incorporation of the downward tilt angle of the camera was inspired by the human perception system that perceives distances by a directional process of integrating ground information up to 20 meters [20]. Indeed, humans cannot judge the absolute distance beyond 2 to 3 meters without these visual cues on ground. Note the two ground lines that define the ground plane of the corridor in Fig. 1.

The concept of the infinity point, (P_x, P_y) was added to obtain vehicle yaw angle and camera pitch angle. The infinity point is an imaginary concept where the projections of the two hallway lines happen to intersect on the image plane. Since this imaginary intersection point is infinitely far from the camera, it presents no parallax from the translation of

the camera. It does, however, effectively represent the yaw and the pitch of the camera. Assume that the end points of the hallway ground lines are $E_{H1} = (l, d, -H)^T$ and $E_{H2} = (l, d-w, -H)^T$ where l is length and w is the width of the hallway, d is the horizontal displacement of the camera from the left wall, and H is the MAV altitude (see Fig. 4 for a visual description). The Euler rotation matrix to convert from the camera frame to the hallway frame is given in (3),

$$A = \begin{bmatrix} c\psi c\beta & c\beta s\psi & -s\beta \\ c\psi s\phi s\beta - c\phi s\psi & c\phi c\psi + s\phi s\psi s\beta & c\beta s\phi \\ s\phi s\psi + c\phi c\psi s\beta & c\phi s\psi s\beta - c\psi s\phi & c\phi c\beta \end{bmatrix} \quad (3)$$

where c and s are abbreviations for cos and sin functions respectively. The vehicle yaw angle is denoted by ψ, the pitch by β, and the roll by ϕ. Since the roll angle is controlled by the onboard autopilot system, it can be set to be zero.

The points E_{H1} and E_{H2} are transformed into the camera frame via multiplication with the transpose of A in (3)

$$E_{C1} = A^T.(l, d, -H)^T, \quad E_{C2} = A^T.(l, d-w, -H)^T \quad (4)$$

This 3D system is then transformed into the 2D image plane via

$$u = yf/x, \quad \text{and} \quad v = zf/x \quad (5)$$

where u is the pixel horizontal position from center (right is positive), v is the pixel vertical position from center (up is positive), and f is the focal length. The end points of the hallway lines have now transformed from $E1_{Hall}$ and $E2_{Hall}$ to $(Px_1, Py_1)^T$ and $(Px_2, Py_2)^T$, respectively. An infinitely long hallway can be represented by

$$\begin{aligned} \lim_{l\to\infty} Px_1 = \lim_{l\to\infty} Px_2 = f\tan\psi \\ \lim_{l\to\infty} Py_1 = \lim_{l\to\infty} Py_2 = -f\tan\beta/\cos\psi \end{aligned} \quad (6)$$

which is conceptually same as extending the hallway lines to infinity. The fact that $Px_1 = Px_2$ and $Py_1 = Py_2$ indicates that the intersection of the lines in the image plane is the end of such an infinitely long hallway. Solving the resulting equations for ψ and β yields the camera yaw and pitch respectively,

$$\psi = \tan^{-1}(P_x/f), \quad \beta = -\tan^{-1}(P_y\cos\psi/f) \quad (7)$$

A generic form of the transformation from the pixel position, (u, v) to (x, y, z), can be derived in a similar fashion. The equations for u and v also provide general coordinates in the camera frame as $(z_c f/v, uz_c/v, z_c)$ where z_c is the z position of the object in the camera frame. Multiplying with (3) transforms the hallway frame coordinates (x, y, z) into functions of u, v, and z_c. Solving the new z equation for z_c and substituting into the equations for x and y yields,

$$\begin{aligned} \tilde{x} &= ((a_{12}u + a_{13}v + a_{11}f)/(a_{32}u + a_{33}v + a_{31}f))z \\ \tilde{y} &= ((a_{22}u + a_{23}v + a_{21}f)/(a_{32}u + a_{33}v + a_{31}f))z \end{aligned} \quad (8)$$

where a_{ij} denotes the elements of the matrix in (3). See Fig. 1 for the descriptions of $\tilde{x}$ and $\tilde{y}$.

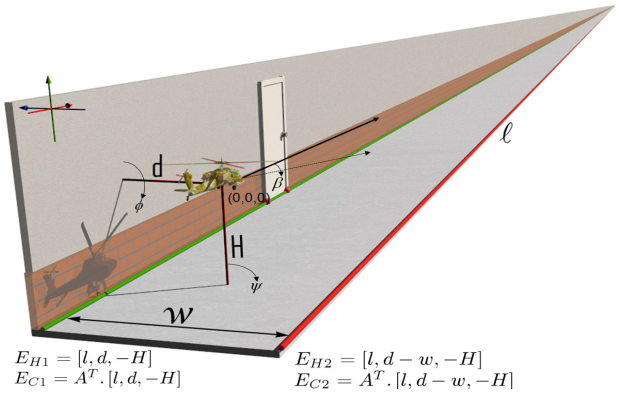

Fig. 4. A visual description the world as perceived by the Infinity-Point Method.

For objects likely to be on the floor, the height of the camera above the ground is the z position of the object. Also, if the platform roll can be measured, or assumed negligible, then the combination of the infinity point with the height can be used to give the range to any object on the floor of the hallway. This same concept applies to objects which are likely to be on the same wall or the ceiling. By exploiting the geometry of the corners present in the corridor, our method computes the absolute range and bearing of the features, effectively turning them into landmarks needed for the SLAM formulation. Our earlier works [15] employed an older method of range measurement, called Line-Perspectives method, which the Infinity-Point method improves in terms of accuracy. However, in the rare event when only one hallway line is detectable, and thus the infinity point is lost, the system switches from the Infinity-Point method to the Line-Perspectives method until both lines are detected again.

III. Helix Bearing Algorithm

In this section, we propose a turn-sensing algorithm to estimate ψ in the absence of orthogonality cues, such as when approaching a turn. This situation automatically triggers the turn-exploration mode in the MAV, in which a yaw rotation of the body frame is initiated until another passage is found. The challenge is to estimate ψ accurately enough to update the SLAM map correctly. This way, the MAV can also determine where turns are located the next time they are visited.

The new measurement problem at turns is to compute the instantaneous velocity, (u, v) of every helix (moving feature) that the MAV is able to detect. In other words, an attempt is made to recover $V(x, y, t) = (u(x, y, t), (v(x, y, t)) = (dx/dt, dy/dt)$ using a variation of the pyramidal Lucas-Kanade method. This recovery leads to a motion field; a 2D vector field obtained via perspective projection of the 3D velocity field of a moving scene onto the image plane. At discrete time steps, the next frame is defined as a function of a previous frame as $I_{t+1}(x, y, z, t) = I_t(x + dx, y + dy, z +$

$dz, t + dt)$. By applying the Taylor series expansion,

$$I(x, y, z, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial z}\delta z + \frac{\partial I}{\partial t}\delta t \qquad (9)$$

then by differentiating with respect to time yields, the helix velocity is obtained in terms of pixel distance per time step k.

At this point, each helix is assumed to be identically distributed and independently positioned on the image plane, associated with a velocity vector $V_i = (v, \varphi)^T$ where φ is the angular displacement of velocity direction from the north of the image plane where $\pi/2$ is east, π is south and $3\pi/2$ is west. Although the associated depths of the helix set appearing at stochastic points on the image plane are unknown, assuming a constant $\dot{\psi}$, there is a relationship between distance of a helix from the camera and its instantaneous velocity on the image plane. This suggests that a helix cluster with respect to closeness of individual instantaneous velocities is likely to belong on the surface of one planar object, such as a door frame. Let a helix with a directional velocity be the triple $h_i = (V_i, u_i, v_i)^T$ where (u_i, v_i) represents the position of this particle on the image plane. At any given time (k), let Ψ be a set containing all these features on the image plane such that $\Psi(k) = \{h_1, h_2, \cdots, h_n\}$. The z component of velocity as obtained in (9) is the determining factor for φ. Since we are most interested in the set of helix in which this component is minimized, $\Psi(k)$ is re-sampled such that,

$$\Psi'(k) = \{\forall h_i, \{\varphi \approx \pi/2\} \cup \{\varphi \approx 3\pi/2\}\} \qquad (10)$$

sorted in increasing velocity order. $\Psi'(k)$ is then processed through histogram sorting to reveal the modal helix set such that,

$$\Psi''(k) = \max \begin{cases} \text{if } (h_i = h_{i+1}), & \sum_{i=0}^{n} i \\ \text{else, } 0 \end{cases} \qquad (11)$$

$\Psi''(k)$ is likely to contain clusters that tend to have a distribution which can be explained by spatial locality with respect to objects in the scene, whereas the rest of the initial helix set from $\Psi(k)$ may not fit this model. The RANSAC algorithm [19] is a useful method to estimate parameters of such models, however for efficiency, an agglomerative hierarchical tree T is used to identify the clusters. To construct the tree, $\Psi''(k)$ is heat mapped, represented as a symmetric matrix M, with respect to Manhattan distance between each individual helix,

$$M = \begin{bmatrix} h_0 - h_0 & \cdots & h_0 - h_n \\ \vdots & \ddots & \vdots \\ h_n - h_0 & \cdots & h_n - h_n \end{bmatrix} \qquad (12)$$

It is desirable to stop the algorithm before it completes since this would eventually result in $\Psi'''(k) = \Psi''(k)$. In other words, the tree should be cut at the sequence m such that $m + 1$ does not provide significant benefit in terms of modeling the clusters. After this step, the set of velocities in $\Psi'''(k)$ represent the largest planar object in the field of view with the most consistent rate of pixel displacement in time. Due to the lack of absolute depth information, if no identifiable objects exist in the field of view, the system is updated such that $\Psi(k+1) = \Psi(k) + \mu(\Psi'''(k))$ as the best effort estimate. However, if the MAV is able to identify a world object of known dimensions, $dim = (x, y)^T$ from its internal object database, such as a door, and the cluster $\Psi'''(k)$ sufficiently coincides with this object, Helix bearing algorithm can estimate depth to this cluster using $dim(f/dim')$ where dim is the actual object dimensions, f is the focal length and dim' represents object dimensions on image plane. Note that the existence of known objects is not required for this method to work, however they would increase its accuracy.

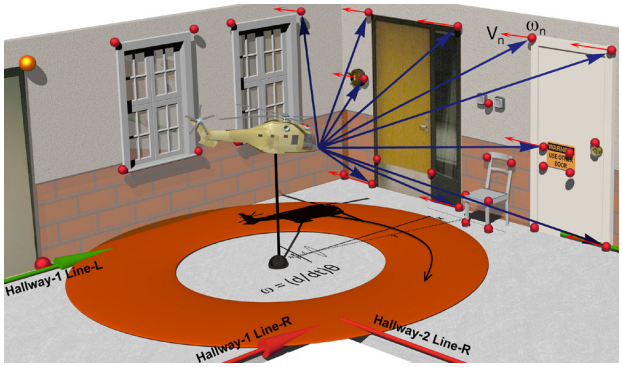

Fig. 5. The Helix bearing algorithm exploits the optical flow field resulting from the features not associated with architectural lines. Helix velocities that form statistically identifiable clusters indicate the presence of planar objects which can help with turn estimation.

IV. SLAM FORMULATION WITH FASTSLAM

Our previous experiments [15] showed that, due to the highly nonlinear nature of the observation equations, traditional nonlinear observers such as EKF do not scale to SLAM in larger environments with vast numbers of potential landmarks. Measurement updates in EKF require quadratic time complexity, rendering the data association increasingly difficult as the map grows. An MAV with limited computational resources is particularly impacted from this complexity behavior. FastSLAM [21] is a dynamic Bayesian approach to SLAM, exploiting the conditional independence of measurements. A random set of particles is generated using the noise model and dynamics of the vehicle in which each particle is considered a potential location for the vehicle. A reduced Kalman filter per particle is then associated with each of the current measurements. Considering the limited computational resources of an MAV, maintaining a set of landmarks large enough to allow for accurate motion estimations, yet sparse enough so as not to produce a negative impact on the system performance is imperative. The noise model of the measurements along with the new measurement and old position of the feature are used to generate a statistical weight. This weight in essence is a measure of how well the landmarks in the previous sensor position correlate with the measured position, taking noise into account. Since each

of the particles has a different estimate of the vehicle position resulting in a different perspective for the measurement, each particle is assigned different weights. Particles are re-sampled every iteration such that the lower weight particles are removed, and higher weight particles are replicated. This results in a cloud of random particles of track towards the best estimation results, which are the positions that yield the best correlation between the previous position of the features, and the new measurement data. The positions of landmarks are stored by the particles such as $Par_n = (X_L^T, P)$ where $X_L = (x_{ci}, y_{ci})$ and P is the 2×2 covariance matrix for the particular Kalman Filter contained by Par_n. The 6DOF vehicle state vector, x_v, can be updated in discrete time steps of (k) as shown in (13) where $R = (x_r, y_r, H)^T$ is the position in inertial frame, from which the velocity in inertial frame can be derived as $\dot{R} = v_E$. The vector $v_B = (v_x, v_y, v_z)^T$ represents linear velocity of the body frame, and $\omega = (p, q, r)^T$ represents the body angular rate. $\Gamma = (\phi, \theta, \psi)^T$ is the Euler angle vector, and L_{EB} is the Euler angle transformation matrix for (ϕ, θ, ψ). The 3×3 matrix T converts $(p, q, r)^T$ to $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. At every step, the MAV is assumed to experience unknown linear and angular accelerations, $V_B = a_B \Delta t$ and $\Omega = \alpha_B \Delta t$ respectively.

$$x_v(k+1) = \begin{pmatrix} R(k) + L_{EB}(\phi, \theta, \psi)(v_B + V_B)\Delta t \\ \Gamma(k) + T(\phi, \theta, \psi)(\omega + \Omega)\Delta t \\ v_B(k) + V_B \\ \omega(k) + \Omega \end{pmatrix} \tag{13}$$

There is only a limited set of orientations a helicopter is capable of sustaining in the air at any given time without partial or complete loss of control. Moreover, the on-board autopilot incorporates IMU and compass measurements in a best-effort scheme to keep the MAV at hover in the absence of external control inputs. Thus, the 6DOF system dynamics in 13 can be simplified into 2D system dynamics with an autopilot, and the MAV can be directed as in 2D car-like mechanics with 180 degree swivel steering.

A. Data Association

As a prerequisite for SLAM to function properly, recently detected landmarks need to be associated with the existing landmarks in the map such that each measurement correspond to the correct landmark. In essence, the association metric depends only on the measurement innovation vector, often leading to data ambiguity in a three dimensional environment. The typical data association method is to compare every measurement with every feature on the map and a measurements becomes associated with a feature if it is sufficiently close to it, a process that would exponentially slow down over time. Moreover, since the measurement is relative, the error of the vehicle position is additive with the absolute location of the measurement. We present a new approach to this issue as a faster and more accurate solution, which takes advantage of landmark locations on the image plane. Landmarks appear to move along the ground lines as the MAV moves, and data association is a problem born from their natural ambiguity. Assume that $p^k_{(x,y)},\ k = 0, 1, 2, 3, \ldots, n$ represents a pixel in time which happens to be contained by a landmark, and this pixel moves along a ground line at the velocity v_p. Although landmarks often contain a cluster of pixels size of which is inversely proportional with landmark distance, here the center pixel of a landmark is referred. Given that the expected maximum velocity, V_{Bmax}, is known, a pixel is expected to appear at

$$p^{k+1}_{(x,y)} = f((p^k_{(x,y)} + (v_B + V_B)\Delta t)) \tag{14}$$

where

$$\sqrt{(p^{k+1}_{(x)} - p^k_{(x)})^2 + (p^{k+1}_{(y)} - p^k_{(y)})^2} \tag{15}$$

cannot be larger than $\frac{V_{Bmax}}{\Delta t}$ and $f(\cdot)$ is a function that converts landmark range to position on the image plane.

A landmark appearing at time $k+1$ is to be associated with a landmark that has appeared at time k if and only if their pixel locations are within the association threshold. In other words, the association information from k is used. Otherwise, if the maximum expected change in pixel location is exceeded, the landmark is considered new. using the association data from k when a match is found instead of searching the large global map saves computational resources. In addition, since the pixel location of a landmark is independent of the noise in the MAV position, the association has an improved accuracy. To further improve accuracy, there is also a maximum range beyond which the MAV will not consider landmarks for data association. This range is determined taking camera resolution into consideration. The farther a landmark is, the fewer pixels it has in its cluster, thus the more ambiguous it becomes and the more noise it may contain. Currently, the MAV is set to ignore landmarks farther than 8 meters.

V. Experimental Results

As illustrated in Fig. 6, our monocular vision SLAM correctly locates and associates landmarks. A 3D map is built by the addition of time-varying altitude and wall-positions, as shown in Fig. 7. In the top-down maps such as Fig. 6, the small circle with a protruding line represents the MAV and its current heading, respectively. Large circles represent landmarks in the process of data association. Circle diameter represents the uncertainty for that landmark position, with larger diameter representing higher uncertainty. At highest level of certainty, the circle becomes invisible. The uncertainty is known in both x and y directions in the inertial frame, therefore these circles are indeed elliptical. However, since the MAV is highly certain about the range of landmarks with respect to distance of walls from each other, the worst of the two uncertainties is used. Also, the diameter of uncertainty is inflated in the figure for visibility. A large uncertainty often represents an inconsistent feature which might have been introduced when external disturbances are present. The proposed methods turn out to be robust to transient disturbances since the corner-like features that might have been introduced by the walking person would have a very high uncertainty, and would not be considered for the map in the long term.

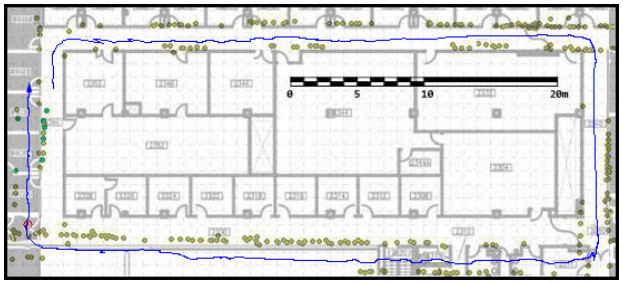

Fig. 6. Experimental results of the proposed ranging and SLAM algorithm. Building floor plan was later superimposed with scale accuracy to provide reference data for the ground truth to demonstrate the performance and accuracy of our method. It is **not** provided to the MAV a-priori.

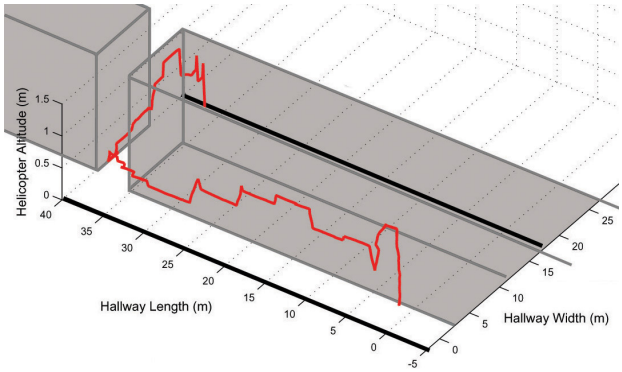

Fig. 7. Cartesian (x, y, z) position of the MAV in a hallway as reported by proposed ranging and SLAM algorithm with time-varying altitude. Altitude was intentionally varied by large amounts to demonstrate the robustness of our method to the climb and descent of the aircraft, whereas in a typical mission natural altitude changes are in the range of a few centimeters.

The MAV assumes that it is positioned at $(0, 0, 0)$ Cartesian coordinates at the start of a mission, with the camera pointed at the positive x axis, therefore, the width of the corridor is represented by the y axis. At anytime during the mission, a partial map can be requested from the MAV. As the MAV features an IEEE 802.11 interface, the map can be requested over an Internet connection as long as the building provides a wireless Internet service or downloaded ad-hoc if a laptop computer is in range. In any case the map is stored in the MAV for later retrieval. The MAV also stores video frames at certain intervals or important events, which are time-linked to the map. It is therefore possible to obtain a still image of the surroundings of any landmark for the surveillance and identification purposes.

In Fig. 6, the loop is over 100 meters. When the system closes the loop, the MAV believes to be within less than 2 meters of where the mission started. It should be stressed that the third leg of this hallway contained no detectable features, considering the MAV ignores landmarks farther than 8 meters. The MAV can still center itself between the walls via the line extraction algorithms. Hence, once the loop is complete, the system is able to quantify the amount of error between the actual starting position and the projected ending position, which can be corrected accordingly in the next iteration of the loop.

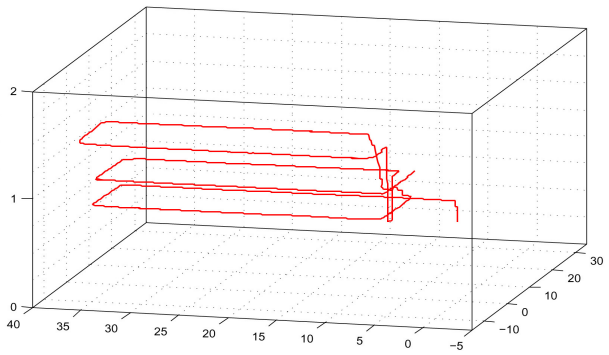

Fig. 8. Cartesian (x, y, z) position of the MAV in a hallway over time, demonstrating the loop-closing performance of the proposed ranging and SLAM algorithm.

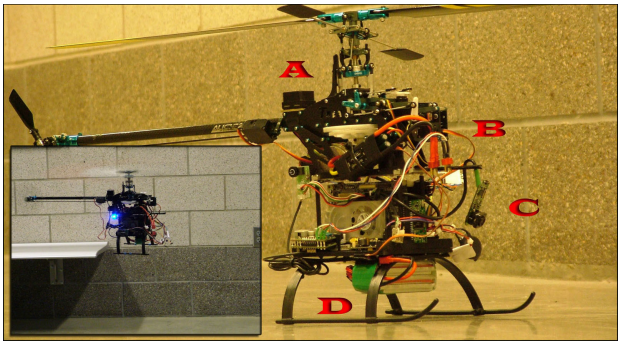

Fig. 9. Saint Vertigo, the autonomous MAV helicopter consists of four decks. The A-deck contains collective pitch rotor head mechanics, The B-deck comprises the fuselage which houses the power-plant, transmission, main batteries, actuators, gyroscope, and the tail rotor. The C-deck is the autopilot compartment which contains the inertial measurement unit, all communication systems, and all sensors. The D-deck carries the navigation computer which is attached to a digital video camera visible at the front.

A. The Micro Aerial Vehicle Hardware Configuration

Saint Vertigo (Fig. 9) is one of the smallest and fully self-contained autonomous helicopters in the world capable of both indoor and outdoor operation. Our unit performs all image processing and SLAM computations on-board via a 1GHz CPU, 1GB RAM, and 4GB mass storage. The MAV can be remotely accessed over a wireless Internet connection. A 900MHz modem and 2.4GHz manual override are included for programming and safety purposes. An additional 2 lbs of payload is available for adaptability to different mission requirements. In essence, the MAV features two independent computers. The *flight computer* is responsible for flight stabilization, flight automation, and sensory management, including but not limited to tracking the time-varying altitude via an ultrasonic altimeter. The *navigation computer* is responsible for higher-consciousness tasks such as image processing, range measurement, SLAM computations, networking, mass-storage, and possibly, path planning. The neural pathway linking them is a dedicated on-board serial communications link, through which the sensory feedback and supervisory commands are shared; straightforward directives which are translated into appro-

TABLE I

CPU UTILIZATION OF THE PROPOSED ALGORITHMS

Image Acquisition and Edge Filtering	10%
Line and Slope Extraction	2%
Landmark Extraction	20%†
Helix Bearing	20%†
Ranging Algorithms	Below 1%
FastSLAM	50%

priate helicopter responses by the flight computer.

B. Processing Requirements

In order to effectively manage the computational resources on a lightweight MAV computer, we keep track of the CPU utilization for the algorithms proposed in this paper. Table I shows a typical breakdown of the average processor utilization per one video frame. Each corresponding task, elucidated in this paper, is visualized in Fig. 2. The numbers in Table I are gathered after the map has matured. Methods highlighted with † are mutually exclusive, e.g., the Helix Bearing algorithm runs only when the MAV is performing turns, while ranging task is on standby. FastSLAM has a roughly constant load on the system once the map is populated. We only consider a limited point cloud with landmarks in the front detection range of the MAV (see Section IV-A). The MAV typically operates at 80% utilization range, with SLAM updates in 15Hz range. It should be stressed that these numerical figures include generic operating system kernel processes, some of which are neither associated with, nor required for the MAV operation. Development of an application-specific operating system for Saint Vertigo is a suggested future goal.

VI. CONCLUSION AND FUTURE WORK

While widely recognized SLAM methods such as FastSLAM have been mainly developed for use with laser range finders, this paper presented new algorithms for monocular vision based depth perception and bearing sensing integrated with 3D SLAM. Our algorithms are shown to be capable of adapting to various situations (e.g., turns, external objects, and time-varying altitude). Further, the proposed monocular vision SLAM method does not need initialization procedures. The system is only limited by the capabilities of the camera all of which can be overcome with the proper use of lenses and higher fidelity imaging sensors. In this study, we have used a consumer-grade USB camera. A purpose-built camera is suggested for future work to allow development of efficient vision SLAM and data association algorithms that take advantage of the intermediate image processing data. Our future vision-based SLAM and navigation strategy for an indoor MAV helicopter through a building also includes the ability to recognize staircases, and thus traverse multiple floors to generate a comprehensive volumetric map of the building. Considering our MAV helicopter is capable of outdoor flight, we can extend our method to the outdoor perimeter of buildings and similar outdoor urban environments by exploiting the similarities between hallways and downtown city maps.

VII. ACKNOWLEDGEMENT

The research reported in this paper was in part supported by National Science Foundation (Grant ECCS-0428040), Information Infrastructure Institute (I^3), Department of Aerospace Engineering and Virtual Reality Application Center at Iowa State University, and Air Force Office of Scientific Research.

REFERENCES

[1] L. M. Paz, P. Pinies, J. D. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946-957, Oct. 008.

[2] H. Andreasson, T. Duckett, A. J. Lilienthal, "A minimalistic approach to appearance-based visual SLAM", *IEEE Trans. Robot.*, vol. 24 pp. 991–1001, Oct. 2008.

[3] N. Isoda, K. Terada, S. Oe, and K. IKaida, "Improvement of accuracy for distance measurement method by using movable CCD," *SICE*, pp. 29–31, Tokushima, July 29-31, 1997.

[4] F. Ruffier, and N. Franceschini, "Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction," *Proc. IEEE Int. Conf. on Robot. and Auto.*, pp. 2339–2346, New Orleans, 2004.

[5] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini, "Bio-inspired optical flow circuits for the visual guidance of micro-air vehicles," *ISCAS*, Bangkok, Thailand, vol. 3, pp. 846–849, May 25-28, 2003.

[6] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Transactions on Graphics*, vol. 25, no. 3, Aug 2006.

[7] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences", *Proc. European Conf. Computer Vision*, pp. 311-326, June 1998.

[8] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters", *Proc. Sixth Int'l Conf. Computer Vision*, pp. 90-96, 1998.

[9] A. Davison, M. Nicholas, and S. Olivier, "MonoSLAM: real-time single camera SLAM," PAMI, vol. 29, no. 6, pp. 1052–1067, 2007.

[10] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos, "Mapping Large Loops with a Single Hand-Held Camera," *In Proc. RSS III*, 2007.

[11] F. Dellaert, W. Burgard, D. Fox, and S. Thrun "Using the CONDENSATION Algorithm for Robust, Vision-based Mobile Robot Localization," *CVPR*, June, 1999.

[12] N. Cuperlier, M. Quoy, P. Gaussier, and C. Giovanangeli, "Navigation and planning in an unknown environment using vision and a cognitive map," *IJCAI Workshop, Reasoning with Uncertainty in Robotics*, 2005.

[13] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM", *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 969–979, Oct. 2008.

[14] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 980–990, Oct. 2008.

[15] K. Çelik, S.-J. Chung, and A. K. Somani, "Mono-vision corner SLAM for indoor navigation". *Proc. IEEE Int'l Conf. on Electro-Information Technology*, Ames, Iowa, May 2008, pp. 343–348.

[16] K. Çelik, S.-J. Chung, and A. K. Somani,"Biologically inspired monocular vision based navigation and mapping in GPS-denied environments," *Proc. AIAA Conf. Infotech at Aerospace and Unmanned Unlimited*, Seattle, WA, 6-9 April 2009.

[17] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. of the 4th. Alvey Vision Conf.*, pp. 147-151, 1988.

[18] J. Shi and C. Tomasi, "Good features to track," *CVPR*, pp. 593-600, June 1994.

[19] D. C. K. Yuen and B. A. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 217–226, Apr. 2005.

[20] B. Wu, T. L. Ooi, and Z. J. He, "Perceiving distance accurately by a directional process of integrating ground information," *Nature*, vol. 428, pp. 73–77, Mar. 2004.

[21] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," *Proc. AAAI Natl. Conf. AI.*, pp. 593–598, 2002.